

**Changes to the Specification**

Please replace paragraph [0018] with the following amended paragraph:

[0018] In one embodiment, the system measures a connection speed to the plurality of servers. The connection speed of the plurality of [[server]] servers is compared to an available bandwidth at the client. This information is used to determine which download system is used. Next it is determined if an available bandwidth is less than a connection speed to two of the servers. When the available bandwidth is less than the connection speed to two of the servers, a serial concatenated download system is selected. A serial concatenated download system selects a first server (one of the list of servers) to start a download. If there is a failure at the first server, a second server is selected. The download is requested to start at a next byte after a last byte received. When the available bandwidth is not less than the connection speed to two of the servers, a multiple concurrent download system is selected. This system starts downloading from multiple servers (at least two servers) at the same time. In another embodiment a multiple concatenated download system is selected when the available bandwidth is not less than the connection speed to two of the servers. The multiple concatenated download system starts a first download at a first byte of the file for one of the at least two servers. Then a second download is started at a second byte of the file for a second of the at least two servers. The files are combined to create a complete file. As will be apparent to those skilled in the art the system can be expanded to n-servers.

Please replace paragraph [0021] with the following amended paragraph:

[0021] FIG. 5 is a schematic diagram of a downloading system in accordance with one embodiment of the invention. This figure shows a multiple concatenated download system. A first server 80 starts downloading a file 82 at a first start byte 84 to the client. Note that the start of file 84 need not be the beginning of the file 86. A second server 88 starts downloading the file 82 at a second start byte 90 to the client. When [[a]] various portions of the file can be combined to form a complete copy of the file, the download is terminated and the portions are concatenated to form the complete file 92. This can be accomplished by having two processes running on the client machine that send a message to each other when an overlapping byte is downloaded or this can be [[down]] done with two threads of the same process. Note that this technique can also be expanded to n-servers.